

Esolang

DiPS CodeJam 24

Prompt

Pranav and Prithvi are back from last year, and this time they're on an adventure to find the secret lost treasure in an ancient Japanese ruin. They realise that the secret to where the treasure lies is hidden in a coded message – one that is represented by some strange symbols. They discover these haikus, corresponding to the structure of the lines (translated to English, of course):

@ & -

*Variable breathes life,
Value whispers in its ear,
New dawn for the name.*

+ & -

*Value joins the dance,
Increasing as shadows grow,
Addition's soft grace.*

* & -

*Strength of numbers blend,
Variable's power grows,
Multiplication.*

/ & -

*Splitting parts with care,
Division by quiet force,
Fair share to the whole.*

-

*Echo of the name,
Whispers truth from within code,
Value in the light.*

!

*End of tales unfolds,
A sudden halt to the quest,
Silence speaks in code.*

Pranav and Prithvi have deciphered these to mean that the coded message is written in some esoteric language, and the syntax is given by the symbols before the haikus. The ampersand represents the name of some variable, and the hyphen represents some numerical value.

For example:

@ x 3 means "Set x to 3"

+ x 4 means "Add 4 to x"

* x 8 means "Multiply x by 8"

/ x 2 means "Divide x by 2"

- x 5 means "Subtract 5 from x"

x means "Print x" (print statements print on new lines and **not** on the same line)

! means "Stop decoding here" (ignore all further statements) and reveals the coded message 23.

Given a message that is n lines long, can you figure out the secret code?

Input Format

- The first line of the input contains an integer n , denoting the number of lines.
- The next n lines of the input each contain the encoded message.

Output Format

Your output should contain the decoded message. If there is no decoded message, type in **no output**.

Sample Program

```
def solve(code):
    code = [c.split() for c in code.split("\n")]

    o = {}
    p = ""

    for c in code:
        if c[0] == "!":
            break

        if c[0] == "@":
            o[c[1]] = int(c[2])

        if c[0] == "+":
            o[c[1]]+=int(c[2])

        if c[0] == "*":
            o[c[1]]*=int(c[2])

        if c[0] == "#":
            p+=f"{o[c[1]]}\n"

    return p[:-1:]
```